International Journal of communication and computer Technologies, ISSN: 2278-9723

Available at http://www.ijccts.org

SECURITY CONFIGURATION AND PERFORMANCE ANALYSIS OF FTP SERVER

[1]Sharad Pratap Singh, [2]Navin Goyal

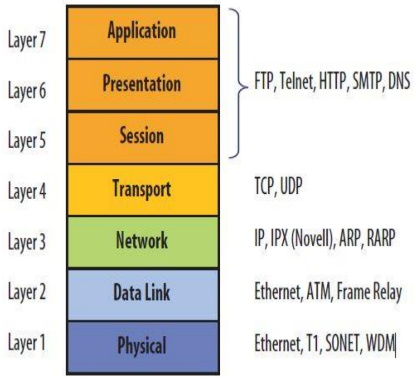
[1,2] Computer Science Engineering, Suresh Gyan Vihar University, Jaipur, India

Received: 02-07-2014, **Revised:** 24-08-2014, **Accepted:** 20-10-2014, **Published online:** 09-12-2014

Abstract—File Transfer Protocol is used for transferring files to clients over networks using client-server architecture. FTP provides two mechanisms for file transfer; one is anonymous method and another one is password authentication mechanism. All the communication between client and server is without encryption means data is transferred in clear text whether it is password or ftp commands. There are some requirements which are considered important while file transfers and these are Authentication, Integrity and Confidentiality. For implementing security in file transfer protocol we use FTPS rather than FTP as it is more secure. As FTPS uses some encryption mechanisms, it adds some extra process which effects the performance. FTP and FTPS are configured according to the security requirements for file transfer. Adding some extra process overhead in FTPS like encryption, it affects the performance. This research paper compares both FTPS and FTP on Linux and Windows server.

1. Introduction

FTP is a widely used application that helps clients to transmit files between computers over a network. There are two parts of FTP server, one is client and another one is server part. The client installs the client application and another laptop runs the server part. The client can upload and download files using a login and password combination and the server checks this data provided by the client for authenticity. Once the server verifies the credentials the client can access the server and complete his work. File Transport Protocol runs on the upper layers of the OSI model. FTP uses layer 4 of OSI model ie, Transport Control Protocol (TCP) to send the data over the network. A TCP connection is initiated through a three-way handshake once the client requests a file from the server.



FTP protocol tells us about how the data can be transferred on to network. It allows user to use remote computers using programs to transfer files over the network. It can also be configured to protect a user from duplications while file is stored among hosts. It helps the clients to send data consistently and efficiently to the server. FTP can be used openly by a user using a terminal but it is designed mainly for use by programs.

1.1 FTPS (File Transfer Protocol Secure)

FTPS protocol adds an extra layer of protection to the clients by implementing encryption and using some secure protocols like TLS protocol to transfer data over the

TCP/IP network. FTPS is the upgraded version of FTP which removes the limitations of the FTP server. FTPS provides secure protocol support as provided by some other services like SMTP (Simple Mail Transfer Protocol Service Extension for Secure SMTP over TLS) and HTTPS (supports Transport Layer Security protocol for secure connection). FTPS overcomes the limitations of FTP server like eavesdrop ping, tampering and message

forgery across the network. It supports full functionality for Secure Socket Layer (SSL) cryptographic protocol and Transport Layer Security protocol. It also implements the use of client-side certificates and server-side public key authentication mechanisms. It also supports well-suited ciphers for transferring data over the network including AES, Triple DES, DES and also some of hash functions such as SHA and MD5.

Typically one of two possible modes is used for FTP over SSL:

- Explicit SSL/TLS –AUTH SSL, AUTH TLS: connection starts on standard FTP port 21, switches to SSL or TLS based on FTP client requesting SSL encryption via AUTH SSL or AUTH TLS command respectively. In Explicit Mode the clients have complete power on which areas of the link are to be encrypted.

- Implicit SSL/TLS –FTP connection starts on a designated port (usually 990), SSL is started at the beginning of the connection. Explicit SSL should be used where standards compliance is mandated. In Implicit Mode, the entire FTP session is encrypted.

FTPS was used in explicit mode in this research.

1.2 FTP Login

The File Transfer Protocol server uses a combination of username and password for the authentication of the clients. The username and password sent over the network using the USER and PASS command. After verifying authentication of the client, the server allow access to the client and then clients can access the files resides on the server. The FTP servers may also be configured as anonymous FTP in which user can access the files on the server without going through any authentication mechanism. For anonymous FTP server we provide “anonymous” for both user name and password. In FTP the communication between the client and the server is in clear text and it makes it vulnerable to many attacks. To avoid these attacks we use FTPS for secure communication.

1.3 Security Issues Associated With FTP

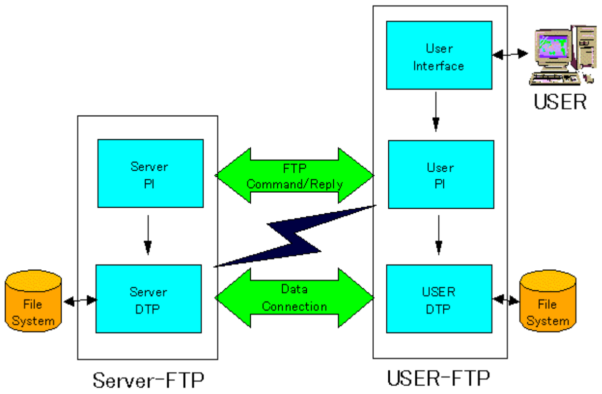
FTP can be used after providing a correct combination of user name and password but the problem arises with this step is that these credentials sent to the server are in clear text format using USER and PASS commands. An attacker can take advantage of the limitation of FTP server sending user credentials in clear text. Attacker can use a sniffer to monitor the network for FTP traffic and read the username and password and got access to the server. Over years FTP protocol facing security related problems such as brute force attack, bounce attacks, spoof attacks, sniffing and port stealing attacks. There are some other protocols which were facing the same problem such as HTTP, SMTP and Telnet. These problems are solved after use of the TLS and SSL protocols. FTP related problems can be solved by using any one of the solution like SSH file transfer protocol (SFTP), Secure Copy (SCP) and FTP with the SSL/TLS (FTPS) to an extent.

1.4 FTP Process Model

FTP protocol works under client-server architecture model. Once the client got authenticated by the server, the server waits for the client side commands to execute the desired action. When client requests the server for ftp link then two transmission channels are opened. One is a control channel (for commands) and another one is for data transmission.
On both the client and the server, there are two processes running which maintain the information flow through the channel.

PI (Protocol Interpreter): The Protocol Interpreter is also divided into two parts i.e. SERVER-PI and CLIENT-PI. The main work of protocol interpreter is to interpret the commands or to understand the commands reside in FTP protocol and perform the desired action.

DTP (Data Transfer Process): The DTP is further divided into two parts one is SERVER-DTP and another is USER-DTP. The main work of data transfer process is the organization of the link between the server and the client.



USER-PI is the first one to communicate with the FTP server. The USER-PI is responsible for connection establishment and also controls the USER-DTP. The transmission of FTP commands and getting replies from the SERVER-PI is also managed by the USER-PI. The commands given by the USER-PI are interpreted by the SERVER-PI over the control channel.
Once the commands are received from the USER-PI, the SERVER-PI comes into action. The SERVER-PI runs the DTP part and opens a port for sending the response to the client. Then the USER-DTP makes a connection to the server on opened port for transferring data.

2. Methodology

The conditions provided for both the windows and Linux systems are nearly same to neglect the differences caused by the various factors while measuring the performances.

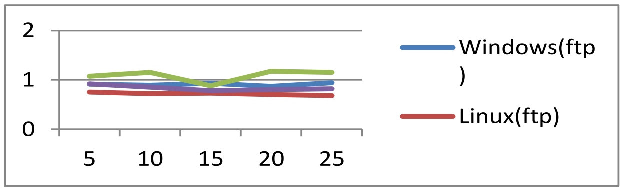
The Computer and the server used are having i5 processor (2.2 GHz) and 3 GB of RAM and connected with client in LAN having 100 Mbps of speed. We use Windows FTP server and Linux FTP server for our research work. For Windows FTP, we use Windows server 2008 r2 and for Linux FTP, we use Red Hat version 6 with kernel version 2.6.32.

The software used to configure the FTP server in RED Hat is vsftpd Very Secure FTP Daemon. Vsftpd is a free, secure FTP server. It supports both FTP and FTPS server.

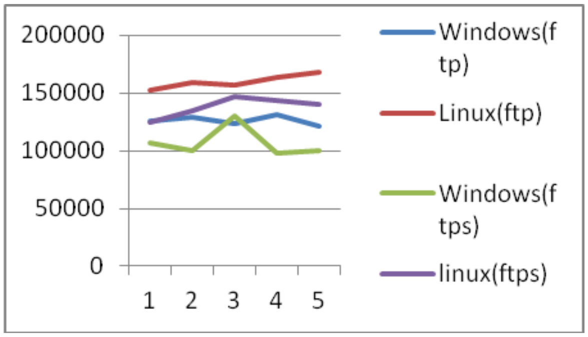
In research work vsftpd and windows FTP server was used as FTP and FTPS server. Windows FTP and vsftpd was used as a FTP server with FTP relevant configurations and as a FTPS server with a self-signed certificate and FTPS relevant configurations. FTPS server was operated only in Explicit Mode.
After Calculating performance for both the servers based on the response time, we also calculate the downloading time taken by the FTP servers for varying file sizes. This step gives a good idea about the working of both the systems.

3. Graphs

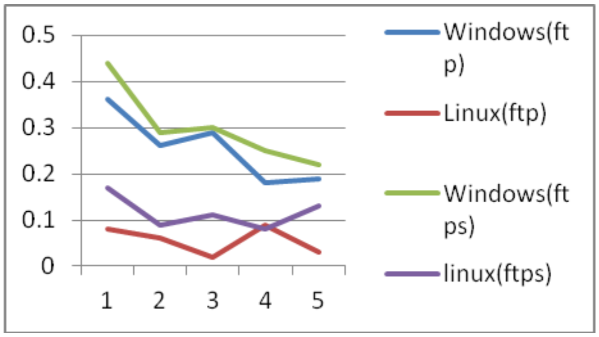
RTT



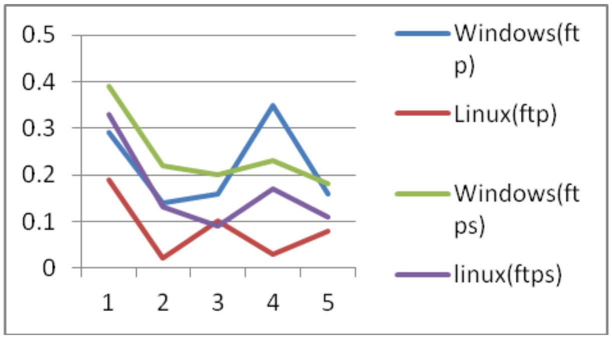
Throughputs



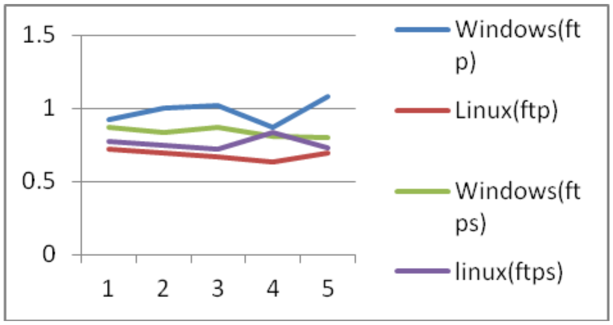
Latency



Jitter



Bandwidth



4. Conclusions

The performance of windows and Linux have been examined under different circumstances like before and after security related modifications and also calculate the time taken by the ftp servers to download the files of various sizes. At the end of the thesis we found the following results.

The RTT, Bandwidth, latency and Jitter for windows is more than the Linux server. This means Linux is faster while doing any operation in ftp server. The Throughput of Linux ftp server is more than that of Windows as a result Linux works faster as it has more capacity to carry data in particular time.

The various parameters during our observation helps in concluding that Linux perform better than windows and can bear heavy load and traffic. With implementation of FTPS the performance of both operating systems depletes slightly. But after implementing FTPS on both operating systems Linux perform better than windows.

There is one more important aspect which is the ease in installing both the operating systems. This method provides a good idea about the knowledge required according to user's viewpoint. Linux server is slightly tough to install and use instead of Windows server.

5. References

[1] Anand Srivastava, “Performance analysis of a Linux based FTP server” 1996.
[2] Dag Henning Liodden Sørbø “Increasing the efficiency of a file server by removing redundant data transfers in popular downloads” 2013.
[3] Roy Gregory Franks, “Performance Analysis of Distributed Server System” 1999.
[4] T. Kiran, “Design and implementation of Transparent Anonymous FTP for Linux” 1998.
[5] J. Postel and J. Reynolds, “File transfer protocol” 1985.
[6] P. Ford-Hutchinson, “Securing FTP with TLS” 2005.
[7] M. Allman and S. Ostermann, “FTP Security Considerations” 1999.
[8] NPradeep Ruwan Nawarathne, ”Overhead of FTP and FTPS over IPsec in IPv6 networks” 2012.